Supplemental File for the Coding of the MC Simulation (SSA Method)

Exposure Models for the Prior Distribution in Bayesian Decision Analysis for Occupational Hygiene Decision Making

by

Eun Gyung Lee,[1] Seung Won Kim,[1*] Charles E. Feigley,[2] and Martin Harper[1]

[1] National Institute for Occupational Safety and Health, Exposure Assessment Branch, Health Effects Laboratory Division, Morgantown, West Virginia
[2] Department of Environmental Health Sciences, Arnold School of Public Health, University of South Carolina Columbia, South Carolina
*Current affiliation: Keimyung University, Daegu, South Korea

Correspondence to: Eun Gyung Lee, NIOSH – HELD, 1095 Willowdale Road, Room 3030, Morgantown, West Virginia; e-mail: dtq5@cdc.gov.

Journal of Occupational and Environmental Hygiene
Volume 10, Issue 2, 2013

Supplemental File for the Coding of the MC Simulation (SSA Method)

```
Sub SSA4pLog()
' Macro coded on 2012.02.02. by Seung Won Kim

    Dim repeat1 As Integer, repeat2 As Integer
    Dim I2min1, I2min2, I2max1, I2max2 As Double ' I1 - intrinsic (task1)
    Dim I3min1, I3min2, I3max1, I3max2 As Double ' I2 - intrinsic (task2)
    Dim H1min1, H1min2, H1max1, H1max2 As Double ' H1 - handling (task1)
    Dim H3min1, H3min2, H3max1, H3max2 As Double ' H3 - handling (task1)

    ' read parameters from the sheet
    repeat1 = Cells(14, 10)
    repeat2 = Cells(15, 10)

    I2min1 = Cells(14, 3)
    I2min2 = Cells(14, 4)
    I2max1 = Cells(14, 5)
    I2max2 = Cells(14, 6)

    I3min1 = Cells(15, 3)
    I3min2 = Cells(15, 4)
    I3max1 = Cells(15, 5)
    I3max2 = Cells(15, 6)

    H1min1 = Cells(16, 3)
    H1min2 = Cells(16, 4)
    H1max1 = Cells(16, 5)
    H1max2 = Cells(16, 6)

    H3min1 = Cells(17, 3)
    H3min2 = Cells(17, 4)
    H3max1 = Cells(17, 5)
    H3max2 = Cells(17, 6)

    ' reset category count
    Cells(20, 3) = 0
    Cells(21, 3) = 0
    Cells(22, 3) = 0
    Cells(23, 3) = 0

    Dim i As Integer, j As Integer, k As Integer, i_95th As Integer
    Dim rI2min, rI2max, rI2 As Double
    Dim rI3min, rI3max, rI3 As Double
    Dim rH1min, rH1max, rH1 As Double
    Dim rH3min, rH3max, rH3 As Double
    Dim C_95th As Double

    ' define one dimension array to find 95th percentile value
    ReDim ArrInnerLoop(1 To repeat1) As Double
```

```
' main loop starts #######################################################################
For j = 1 To repeat2

   ' MC minimum of int(intrinsic emission) between I2min1 and I2min2
   rI2min = Exp(Log(I2min1) + (Log(I2min2) - Log(I2min1)) * Rnd())
   ' MC maximum of int between I2max1 and I2max2
   If rI2min > I2max1 Then
   ' preventing min > max
      rI2max = Exp(Log(rI2min) + (Log(I2max2) - Log(rI2min)) * Rnd())
   Else
      rI2max = Exp(Log(I2max1) + (Log(I2max2) - Log(I2max1)) * Rnd())
   End If

   ' MC minimum of int(intrinsic emission) between I3min1 and I3min2
   rI3min = Exp(Log(I3min1) + (Log(I3min2) - Log(I3min1)) * Rnd())
   ' MC maximum of int between I3max1 and I3max2
   If rI3min > I3max1 Then
   ' preventing min > max
      rI3max = Exp(Log(rI3min) + (Log(I3max2) - Log(rI3min)) * Rnd())
   Else
      rI3max = Exp(Log(I3max1) + (Log(I3max2) - Log(I3max1)) * Rnd())
   End If

   ' MC minimum of handling between H1min1 and H1min2
   rH1min = Exp(Log(H1min1) + (Log(H1min2) - Log(H1min1)) * Rnd())
   ' MC maximum of int between H1max1 and H1max2
   If rH1min > H1max1 Then
   ' preventing min > max
      rH1max = Exp(Log(rH1min) + (Log(H1max2) - Log(rH1min)) * Rnd())
   Else
      rH1max = Exp(Log(H1max1) + (Log(H1max2) - Log(H1max1)) * Rnd())
   End If

   ' MC minimum of handling between H3min1 and H3min2
   rH3min = Exp(Log(H3min1) + (Log(H3min2) - Log(H3min1)) * Rnd())
   ' MC maximum of int between H3max1 and H3max2
   If rH3min > H3max1 Then
   ' preventing min > max
      rH3max = Exp(Log(rH3min) + (Log(H3max2) - Log(rH3min)) * Rnd())
   Else
      rH3max = Exp(Log(H3max1) + (Log(H3max2) - Log(H3max1)) * Rnd())
   End If

   For i = 1 To repeat1

      ArrInnerLoop(i) = 0

      ' MC intrinsic emission between min and max
      rI2 = Exp(Log(rI2min) + (Log(rI2max) - Log(rI2min)) * Rnd())
      Cells(14, 2) = rI2
```

```
         rI3 = Exp(Log(rI3min) + (Log(rI3max) - Log(rI3min)) * Rnd())
         Cells(15, 2) = rI3

         rH1 = Exp(Log(rH1min) + (Log(rH1max) - Log(rH1min)) * Rnd())
         Cells(16, 2) = rH1

         rH3 = Exp(Log(rH3min) + (Log(rH3max) - Log(rH3min)) * Rnd())
         Cells(17, 2) = rH3

         ' fill the array
         ArrInnerLoop(i) = Cells(11, 11)

      Next i

      ' sort the array
      Dim lLoop, lLoop2 As Integer
      Dim NormC1, NormC2 As Double

      For lLoop = 1 To UBound(ArrInnerLoop)
         For lLoop2 = lLoop To UBound(ArrInnerLoop)
            If ArrInnerLoop(lLoop2) < ArrInnerLoop(lLoop) Then
               NormC1 = ArrInnerLoop(lLoop)
               NormC2 = ArrInnerLoop(lLoop2)
               ArrInnerLoop(lLoop) = NormC2
               ArrInnerLoop(lLoop2) = NormC1
            End If
         Next lLoop2
      Next lLoop

      ' find 95th percentile value
      i_95th = WorksheetFunction.Round(repeat1 * 0.95, 0)
      C_95th = ArrInnerLoop(i_95th)

      ' decide and add up its category
      If C_95th < Cells(21, 2) Then
         Cells(20, 3) = Cells(20, 3) + 1
      ElseIf C_95th >= Cells(21, 2) And C_95th < Cells(22, 2) Then
         Cells(21, 3) = Cells(21, 3) + 1
      ElseIf C_95th >= Cells(22, 2) And C_95th < Cells(23, 2) Then
         Cells(22, 3) = Cells(22, 3) + 1
      ElseIf C_95th > Cells(23, 2) Then
         Cells(23, 3) = Cells(23, 3) + 1
      End If

   ' repeat this repeat2 times
   Next j
' main loop ends here ######################################################

End Sub
```